NoνA Simulations technical details

Leon Mualem, University of Minnesota
8/27/2005

The simulation is based on GMINOS and is configured with only input cards (.ffr) files. My personal parts and a lot of scripts are located in
/afs/fnal.gov/files/data/minos/d13/mualem/gminos/

I would recommend just taking a copy of this directory that you can modify and delete as needed.

There are references in MINOS NUMI notes and some ancient web pages for the GMINOS system:

http://www-numi.fnal.gov/numinotes/public/ps/numi0479/numi0479.ps.gz
http://mimosa.astro.indiana.edu/minos/simulation/user_geom.html

There were many geometric "versions" of the detector with different cell sizes and plane configurations. The current standard is called "C". This ends up in a lot of file names, many are called tavc_... indicating it is a totally active detector "ta", and version c, "vc". The differences between versions are primarily cell sizes, and version c is the one with 3.87cm x 6.0 cm inside dimensions.

Here is a table of the simulation parameters for NOvA, the letter in the column is the "version" template, where filenames are usually written as tavX..., and X is the geometric version. A dash (-) in a cell indicates that this has not been simulated, and no template exists for it.

		Width (cm)			
		3.87	5.23	7.95	12.05
	4.5	A	B	E	-
	6.0	C	D	-	I
Depth(cm)	9.0	F	-	G	-
	15.0	H	-	-	J

The basic idea is that there is a template file that defines the geometry, tavc_templ.ffr for example. The script will take that file and add any additional cards needed, defining the number of triggers, the particle type, energy if needed and write it all to a separate .ffr file that will be used for a particular gminos run. Then I start a job that actually runs.

There are some additional files that are called *.fiberinfo that have the definition of fiber pigtails. This is not very important, but needed for everything to work correctly. There is some version of attenuation length differences in there, and some light level assumed, but I am not entirely sure what it is. I assume at first you will just use MC truth energy, and we can develop some code to do smearing of the signal using that APD response as things get more developed.

In general the output files are large, and therefore I usually use some sort of /usr/scratch space at fnal, and copy file to other space as needed.

An example might help.

Running gminos to get an output file:

There is a file called runbr.sh. This takes three arguments, an angle (actually cos(theta) of a zenith angle), and run, which is used a run identifier and random seed, and a background source. Currently the source can be photon or neutron. You probably want something with muon too.

So you might call:
runbr.sh 20 215 photon
This would eventually produce some files called ta_photon20215...

One file that is produced along the way is called ta_photon20215.inp, is produced by CRphotons.C, and is a series of input "events" that will be run. It sets up the input particle with a KINE card, and runs it with a TRIG call.

The output file goes to /usr/scratch/sect1/ta_photon20215.fz_gaf

Once the file is done, there is a rerootjob run on the the file to produce a /usr/scratch/sect1/ta_photon20215.root This can be done for any gaf file produced from gminos and simply translates the tables into root classes and puts them in an event tree.

Reconstruction:

Once you have produced a data file you need to run reconstruction. There are 2 versions of reconstruction that are run on all the neutrino events. One is called “Stan’s”, and the other is called “Peter’s”, for Stan Wojcicki and Peter Litchfield respectively. They each reside in their own subdirectory called *stan* for Stan’s code, and *theseus* for Peter’s code.

The output of each of these codes is an HBOOK ntuple. The analysis of these ntuples is done by separate code, talk to Peter or Stan to get it. Stan’s analysis runs on a root version of the ntuple produced by the program h2root, while Peter’s runs on the hbook ntuple directly.

The Stan ntuple files are located in /afs/fnal.gov/files/data/minos/d56/mualem/new/, and are named with an extension .ntup.root.

Looking at events:

Once this is done, you might want to look at the events. This is done with a display program called AutoDisplay.C, which runs under loon to get some of the libraries. There are actually very few that you need, but it is probably the easiest.

From a loon prompt:

.x AutoDisplay.C("FILENAME",EventNum,ShowTruth,ShowPlots);

FILENAME is the .root file

EventNum is the event to get from the file. If EventNum=-1 it will start at 0 and continue displaying the next event after a carriage return.

ShowTruth turns the display of truth tracks on and off, 0 is off, anything else is on. Will default to on.

ShowPlots turns on/off the display of the plots altogether. Defaults to on if omitted, 0 will turn off.

Truth tracks:
Black proton
Magenta neutron.
Red CC lepton; electron or mu
Green Gamma or pi0,
Blue pi+,pi-

The Display just uses the truth energy deposits, we should move to one that has actual pulse height values, but I think this is a good place to start at least.